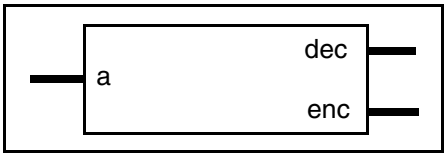
DW_lsd

Leading Signs Detector

Version, STAR, and myDesignWare Subscriptions: IP Directory

Features and Benefits

- Parameterized word length
- Inferable using a function call



Description

DW_lsd contains two outputs, `dec` and `enc`. The `dec` output is a decoded one-hot value of the `a` input vector with a "1" at the least significant (right-most) sign bit position of `a`. The least significant sign bit is the first bit starting from the most significant bit that differs from the next lower bit. The bits from this position up to the most significant bit are all equivalent sign bits, that is input `a` is sign-extended. The output `enc` represents the number of extended sign bits found (from the most significant bit) before the least significant sign bit of input `a`. All lower order bits (to the right) from the first occurrence of a bit that differs from the sign bit are "don't care."

The output port `enc` width is automatically derived from the input port width parameter, `a_width`, and is defined as ceil($\log_2$[*a_width*]) as listed in Table 1-1 The output port `dec` has the same width as the `a` input.

Table 1-1 Pin Description

Pin Name	Width	Direction	Function
a	*a_width*	Input	Input vector
enc	ceil($\log_2$[*a_width*])	Output	Number of leading sign bits in input `a` before the least significant sign bit
dec	*a_width*	Output	One-hot decode of input `a`

Table 1-2 Parameter Description

Parameter	Values	Description
a_width	≤ 1 Default: 8	Vector width of input `a`

Table 1-3 Synthesis Implementations

Implementation Name	Function	License Feature Required
str	Synthesis model	DesignWare

Table 1-4 Simulation Models

Model	Function
DW01.DW_LSD_CFG_SIM	Design unit name for VHDL simulation
dw/dw01/src/DW_lsd_sim.vhd	VHDL simulation model source code
dw/sim_ver/DW_lsd.v	Verilog simulation model source code

Table 1-5 Truth Table (a_width = 8, dec width = 8, enc width = 3)

a(7:0)[a]								enc(2:0)	dec(7:0)							
S	S	S	S	S	S	S	S	111	0	0	0	0	0	0	0	1
S	S	S	S	S	S	S	$\overline{S}$	110	0	0	0	0	0	0	1	0
S	S	S	S	S	S	$\overline{S}$	X	101	0	0	0	0	0	1	0	0
S	S	S	S	S	$\overline{S}$	X	X	100	0	0	0	0	1	0	0	0
S	S	S	S	$\overline{S}$	X	X	X	011	0	0	0	1	0	0	0	0
S	S	S	$\overline{S}$	X	X	X	X	010	0	0	1	0	0	0	0	0
S	S	$\overline{S}$	X	X	X	X	X	001	0	1	0	0	0	0	0	0
S	$\overline{S}$	X	X	X	X	X	X	000	1	0	0	0	0	0	0	0

a. “S” denotes the sign bit or bits that are equivalent to it (sign-extended bits)
“$\overline{S}$” denotes a bit that is different from the sign bit (the inverse of the sign bit)

Related Topics

- Math – Arithmetic Overview
- DesignWare Building Block IP User Guide

HDL Usage Through Function Inferencing - VHDL

```
library IEEE, DWARE;
use IEEE.std_logic_1164.all;
use DWARE.DW_Foundation.all;
-- If using library std_logic_arith, comment line above and
--  uncomment line below
-- use DWARE.DW_Foundation_arith.all;


entity DW_lsd_func is

  generic (
    func_a_width : POSITIVE := 8);

  port (
    func_a   : in  std_logic_vector(func_a_width-1 downto 0);
    enc_func : out std_logic_vector(bit_width(func_a_width)-1 downto 0);
    dec_func : out std_logic_vector(func_a_width-1 downto 0));

end DW_lsd_func;


architecture func of DW_lsd_func is

begin

    -- function inference of DW_lsd
    enc_func <= DWF_lsd_enc (func_a);
    dec_func <= DWF_lsd (func_a);

end func;


-- pragma translate_off
configuration DW_lsd_func_cfg_func of DW_lsd_func is
  for func
  end for;
end DW_lsd_func_cfg_func;
-- pragma translate_on
```

HDL Usage Through Function Inferencing - Verilog

```
module DW_lsd_func (func_a, enc_func, dec_func);

  parameter func_a_width = 8;

`define enc_width 3// enc_width is set to ceil(log2(a_width))

  // Passes the widths to DW_lsd_function
  parameter a_width     = func_a_width;
  parameter addr_width  = `enc_width;

`include "DW_lsd_function.inc"

  input  [func_a_width-1 : 0] func_a;
  output   [`enc_width-1 : 0] enc_func;
  output [func_a_width-1 : 0] dec_func;

  // Function inference of DW_lsd_enc and DW_lsd
  assign enc_func = DWF_lsd_enc (func_a);
  assign dec_func = DWF_lsd (func_a);

endmodule
```

HDL Usage Through Component Instantiation - VHDL

```
library IEEE,DWARE;
use IEEE.std_logic_1164.all;
use DWARE.DWpackages.all;
use DWARE.DW_Foundation_comp.all;

entity DW_lsd_inst is
      generic (
        inst_a_width : POSITIVE := 8
        );
      port (
        inst_a : in std_logic_vector(inst_a_width-1 downto 0);
        enc_inst : out std_logic_vector(bit_width(inst_a_width)-1 downto 0);
        dec_inst : out std_logic_vector(inst_a_width-1 downto 0)
        );
    end DW_lsd_inst;


architecture inst of DW_lsd_inst is

begin

    -- Instance of DW_lsd
    U1 : DW_lsd
    generic map (
          a_width => inst_a_width
          )
    port map (
          a => inst_a,
          enc => enc_inst,
          dec => dec_inst
          );


end inst;
```

HDL Usage Through Component Instantiation - Verilog

```
module DW_lsd_inst( inst_a, enc_inst, dec_inst );

parameter inst_a_width = 8;

`define bit_width_a_width 3// set to ceil(log2(a_width))

input [inst_a_width-1 : 0] inst_a;
output [`bit_width_a_width-1 : 0] enc_inst;
output [inst_a_width-1 : 0] dec_inst;

    // Instance of DW_lsd
    DW_lsd #(inst_a_width) U1 (
                .a(inst_a),
                .enc(enc_inst),
                .dec(dec_inst) );

endmodule
```

Copyright Notice and Proprietary Information

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at https://www.synopsys.com/company/legal/trademarks-brands.html.

All other product or company names may be trademarks of their respective owners.

Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.
www.synopsys.com